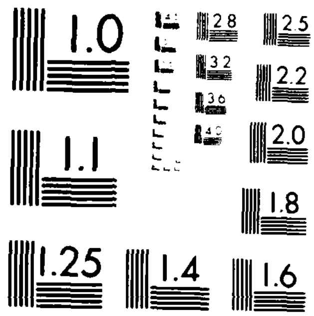
AD-A122 850 THE COMMON THREAD IN OPTIMAL ADAPTIVE NON RECURSIVE FILTERS(U) SAN DIEGO STATE UNIV CA DEPT OF ELECTRICAL AND COMPUTER ENGINEERING F J HARRIS 04 AUG 81 1/1

UNCLASSIFIED N00014-77-C-0260 F/G 12/1 NL

END
DATE
FILMED
[illegible]
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(6)

AD A122850

THE COMMON THREAD IN OPTIMAL ADAPTIVE NON RECURSIVE FILTERS

by

fredric j harris

PROFESSOR

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SAN DIEGO STATE UNIVERSITY

CONTRACT # N00014-77-C-0260

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) San Diego State University Foundation San Diego State University San Diego, CA 92182	2a. REPORT SECURITY CLASSIFICATION AD A12L 8 3 2b. GROUP

3. REPORT TITLE

The Common Thread in Optimal Adaptive Non Recursive Filters

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Final Technical Report, April 1, 1977 through September 30, 1978

5. AUTHOR(S) (First name, middle initial, last name)

Frederick J. Harris, Professor, Department of Electrical and Computer Engineering, San Diego State University.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 4, 1981	38	

8a. CONTRACT OR GRANT NO. N00014-77-C-0260 b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research Mathematical & Information Sciences Div. 800 N. Quincy St. Arlington, Virginia 22217

13. ABSTRACT

Filters are applied to time sequences to extract desired signals from random noise and from interfering signals. The specific form of the filter is related to apriori knowledge of the statistics of the desired and the undesired signals. Classic filter design entails two distinct stages. The first is to employ some monitoring and smoothing scheme which will lead to estimates of the signal statistics such as the signal and noise covariance functions or power spectrums. The second stage is to formulate the filter response in terms of these estimated statistics.

In this paper we review a class of filters for which the two stages just described are performed simultaneously and iteratively. The filter coefficients are changed by a recursive algorithm which corrects the filter response during the processing of the input data. The capability to modify the filter response during operation makes it possible to track and to filter signals with slowly changing statistics.

DD FORM 1 NOV 65 1473

Security Classification

OPTIMAL ADAPTIVE NON RECURSIVE FILTERS

I. INTRODUCTION:

Filters are applied to time sequences to extract desired signals from random noise and from interfering signals. The specific form of the filter is related to apriori knowledge of the statistics of the desired and the undesired signals. Classic filter design entails two distinct stages. The first is to employ some monitoring and smoothing scheme which will lead to estimates of the signal statistics such as the signal and noise covariance functions or power spectrums. The second stage is to formulate the filter response in terms of these estimated statistics.

In this paper we review a class of filters for which the two stages just described are performed simultaneously and iteratively. The filter coefficients are changed by a recursive algorithm which corrects the filter response during the processing of the input data. The capability to modify the filter response during operation makes it possible to track and to filter signals with slowly changing statistics.

II. BACKGROUND:

A discrete non recursive filter is one for which the present output y(k) is obtained as a weighted summation of the past, present, and future inputs x(k+j) with $-N \leq j \leq +N$. The output of such a filter is simply the finite convolution sum shown in Eq (1).

$$y(k) = \sum_{j=-N}^{N} w_j \; x(k-j) \tag{1}$$

For realizability, the index of summation is restricted to be non negative as shown in Eq (2).

$$y(k) = \sum_{j=0}^{N} w_j \; x(k-j) \tag{2}$$

A block diagram of such a filter is shown in figure 1.

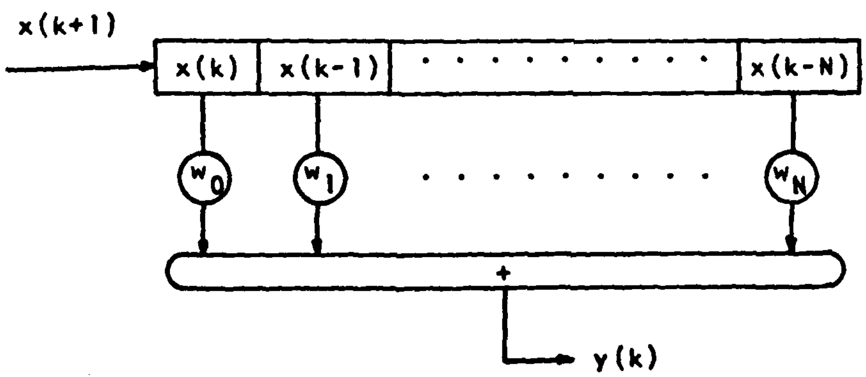


FIGURE 1. A REALIZABLE NON RECURSIVE FILTER

A time varying non recursive filter is one for which the weighting coefficients w_j are allowed to change in some prescribed manner prior to each filter computation. Thus the w_j's are functions of the time index as well as the position index and we will use the notation $w_j(k)$. This is shown in Eq (3).

$$y(k) = \sum_{j=0}^{N} w_j(k)\ x(k-j) \tag{3}$$

For simplicity of notation, let us define $\overline{W}(k)$ as the N+1 dimensional column vector of weights at time k, and $\overline{X}(k)$ as the N+1 dimensional column vector of data at time k. These vectors are shown in Eq (4)

$$\overline{W}(k) = \begin{bmatrix} w_0(k) \\ w_1(k) \\ w_2(k) \\ \cdot \\ \cdot \\ \cdot \\ w_N(k) \end{bmatrix} \qquad \overline{X}(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ x(k-2) \\ \cdot \\ \cdot \\ \cdot \\ x(k-N) \end{bmatrix} \tag{4}$$

Then Eq (3) defining y(k) can be written as a simple inner-product as indicated in Eq (5), where of course, the superscript

$$y(k) = \overline{W}^T(k)\ \overline{X}(k) = \overline{X}^T(k)\ \overline{W}(k) \tag{5}$$

"T" indicates the transpose of the vector.

An adaptive non recursive filter is one for which the weight vector $\overline{W}(k)$ is computed by an algorithm to reduce a cost function. This cost function compares the output of the filter y(k) to an auxiliary input v(k) and is classically selected to be a quadratic function of the difference.

it is precisely this cost function and the corresponding algorithms that this paper addresses.

III THE NORMAL EQUATION.

The model of the filter we are examining is shown in Figure 2.

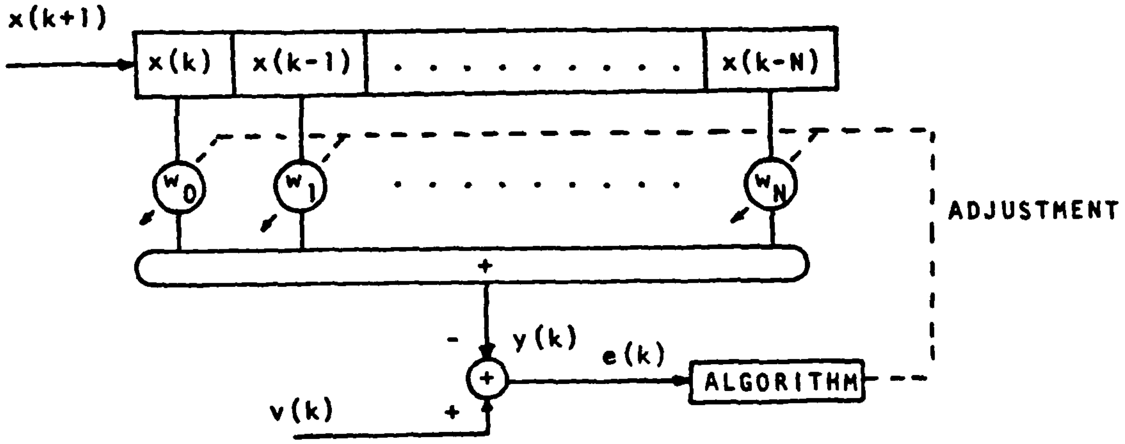


FIGURE 2: ADAPTIVE NON RECURSIVE FILTER

From Figure 2. we define the error at time k, e(k) by Eq(6).

$$e(k) = v(k) - y(k) \tag{6a}$$

$$= v(k) - \overline{W}^T(k)\ \overline{X}(k) \tag{6b}$$

The square of this error $e^2(k)$ is shown in Eq(7).

$$e^2(k) = [v(k) - \overline{W}^T(k)\ \overline{X}(k)]^2 \tag{7a}$$

$$= v^2(k) - 2\ v(k)\ \overline{X}^T(k)\overline{W}(k) + \overline{W}^T(k)\overline{X}(k)\overline{X}^T(k)\overline{W}(k) \tag{7b}$$

The expected value of this squared error (or mean square error) is shown in Eq (8).

$$E\{e^2(k)\} = R_{vv}(k) - 2\overline{R}_{vx}^T(k)\overline{W}(k) + \overline{W}^T(k)R_{\overline{x}\overline{x}}(k)\overline{W}(k) \qquad (8)$$

Where $\overline{R}_{vx}(k)$ is the correlation vector between the desired signal $v(k)$ and the data vector $\overline{X}(k)$ as shown in Eq(9),

$$\overline{R}_{vx}(k) = E\{v(k)\overline{X}(k)\} = \begin{bmatrix} E\{v(k)x(k)\} \\ E\{v(k)x(k-1)\} \\ E\{v(k)x(k-2)\} \\ \cdot \\ \cdot \\ \cdot \\ E\{v(k)x(k-N)\} \end{bmatrix} = \begin{bmatrix} R_{vx}(0) \\ R_{vx}(1) \\ R_{vx}(2) \\ \\ \\ \\ R_{vx}(N) \end{bmatrix} \qquad (9)$$

and where R(k) is the covariance matrix of the input data vector as shown in Eq (10),

$$R_{\overline{x}\overline{x}}(k) = E\{\overline{X}(k)\overline{X}^T(k)\} \qquad (10a)$$

$$= E \begin{bmatrix} x(k)x(k) & x(k)x(k-1) & \cdots & x(k)x(k-N) \\ x(k-1)x(k) & x(k-1)x(k-1) & \cdots & x(k-1)x(k-N) \\ x(k-2)x(k) & x(k-2)x(k-1) & \cdots & x(k-2)x(k-N) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ x(k-N)x(k) & x(k-N)x(k-1) & \cdots & x(k-N)x(k-N) \end{bmatrix} \qquad (10b)$$

$$= \begin{bmatrix} R_{xx}(0) & R_{xx}(-1) & R_{xx}(-2) & \cdots & R_{xx}(-N) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(-1) & \cdots & R_{xx}(1-N) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(2-N) \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ R_{xx}(N) & R_{xx}(N-1) & R_{xx}(N-2) & \cdots & R_{xx}(0) \end{bmatrix} \qquad (10c)$$

and where $R_{vv}(k)$ is the covariance of the desired signal $v(k)$.

We wish to minimize this mean squared error with respect to the the weight vector $\overline{W}(k)$. We accomplish this by computing the first variation (or the gradient) of the squared error with respect to the weight vector $\overline{W}(k)$ and then setting this first variation to zero. This is shown in Eq (11).

$$\delta E\{e^2(k)\} = -2\overline{R}_{vx}(k)\delta\overline{W}(k) + \delta\overline{W}^T(k)R_{\bar{x}\bar{x}}\overline{W}(k) + \overline{W}^T(k)R_{\bar{x}\bar{x}}\delta\overline{W}(k) \tag{11a}$$

$$= -2\ W^T(k)[R_{vx}(k) - R_{\bar{x}\bar{x}}\overline{W}(k)] \tag{11b}$$

The first variation is zero if

$$\overline{R}_{vx}(k) - R_{\bar{x}\bar{x}}\overline{W}(k) = 0 \tag{11c}$$

We note that Eq(11c) is called the Normal Equation, and the solution $\overline{W}(k)$ which minimizes the mean squared error can be obtained by inverting the data covariance matrix $R_{\bar{x}\bar{x}}$. The resultant solution shown in Eq(12) is the Weiner-Hopf solution in matrix form.

$$W(k) = R_{\bar{x}\bar{x}}^{-1}\overline{R}_{vx}(k) \tag{12}$$

Equation (11c) can also be solved iteratively by gradient descent techniques which we will now examine.

IV. GRADIENT DESCENT TO NORMAL SOLUTION:

We define at time k, the i-th iterative approximation to $\overline{W}(k)$ by $\overline{W}_i(k)$. We also define at time k, the i-th residual $\overline{r}_i(k)$ obtained by trying $\overline{W}_i(k)$ in the normal equation. This is shown in Eq (13).

$$\overline{r}_i(k) = \overline{R}_{vx}(k) - R_{\bar{x}\bar{x}}\overline{W}_i(k) \tag{13}$$

We simply seek a technique which will, at each iteration, reduce the residual $\bar{r}_i(k)$. We note that the residual $\bar{r}_i(k)$ is half the gradient vector of the mean squared error evaluated at $\bar{W}_i(k)$, see Eq(11b). Of course reducing the residual to zero is same as reducing the gradient to zero which is the first necessary condition for an unconstrained local extrema. Since the mean squared error is simply quadratic in the weight vector, there is only one extrema which means that the local extrema corresponds to the optimal weight vector.

The iterative correction technique is simply to change the weight vector $\bar{W}_i(k)$ in the direction of the gradient. The gradient point down hill towards the local extrema. The change in the weight vector is shown in Eq (14).

$$\bar{W}_{i+1}(k) = \bar{W}_i(k) + \alpha_i(k)\,\bar{r}_i(k) \tag{14}$$

The scalar α_i controls the rate of convergence of the algorithm by establishing the size of the step in the *direction of the* gradient. It is also simple to bound the step size to assure convergence. This will be done later. For now we will derive the value of α which maximizes the rate of convergence. When this optimal value of α is used in the gradient descent algorithm, the algorithm is called steepest descent. We determine the optimal α by substituting Eq (14) into Eq (8). We obtain Eq (15).

$$\begin{aligned} E\{e^2(k)\} = R_{vv}(k) &- 2\bar{R}^T_{vx}(k)[\bar{W}_i(k)+\alpha_i(k)\bar{r}_i(k)] \\ &+[\bar{W}_i(k)+\alpha_i(k)\bar{r}_i(k)]^T R_{xx}[\bar{W}_i(k)+\alpha_i(k)\bar{r}_i(k)] \end{aligned} \tag{15}$$

We now take the first variation with respect to $\alpha_i(k)$ and obtain Eq (16).

$$\delta E\{e^2(k)\} = -2\delta\alpha_i(k)[\bar{R}^T_{vx}(k)\bar{r}_i(k)]$$

$$+\delta\alpha_i(k)\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{W}_i(k)$$

$$+\bar{W}_i(k)R_{\bar{X}\bar{X}}\delta\alpha_i(k)\bar{r}_i(k)$$

$$+2\delta\alpha_i(k)\alpha_i(k)\bar{r}_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k) \qquad (16a)$$

$$= -2\delta\alpha_i(k)[\bar{R}^T_{vx}(k)\bar{r}_i(k)-\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{W}_i(k)$$

$$-\alpha_i(k)\bar{r}_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k)] \qquad (16b)$$

$$= -2\delta\alpha_i(k)\{\bar{r}^T_i(k)[\bar{R}_{vx}(k)-R_{\bar{X}\bar{X}}\bar{W}_i(k)]$$

$$-\alpha_i(k)[\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k)]\} \qquad (16c)$$

$$= -2\delta\alpha_i(k)\{\bar{r}^T_i(k)\bar{r}_i(k)-\alpha_i(k)\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k)\} \qquad (16d)$$

The first variation is zero if Eq (17) holds.

$$\bar{r}^T_i(k)\bar{r}_i(k) - \alpha_i(k)[\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k)] = 0 \qquad (17a)$$

$$\text{or;} \quad \alpha_i(k) = \frac{\bar{r}^T_i(k)\bar{r}_i(k)}{\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k)} \qquad (17b)$$

The total algorithm for the method of steepest descent is presented in Eq (18).

$$\bar{r}_i(k) = \bar{R}_{vx}(k) - R_{\bar{X}\bar{X}}\bar{W}_i(k) \qquad (18a)$$

$$\alpha_i(k) = \frac{\bar{r}^T_i(k)\bar{r}_i(k)}{\bar{r}^T_i(k)R_{\bar{X}\bar{X}}\bar{r}_i(k)} \qquad (18b)$$

$$\bar{W}_{i+1}(k) = \bar{W}_i(k) + \alpha_i(k)\bar{r}_i(k) \qquad (18c)$$

Note that the index of iteration of the steepest descent is "i", and that the index "k" is the one which evolves with time to allow for time varying statistics. Also note that if the constant $\alpha_i(k)$ is selected to be a constant $\alpha(k)$, the method is simply a gradient descent (as opposed to a steepest descent).

The gradient algorithm can be described by a signal flow graph. The graph presents the inter-relations of the algorithm in a clear concise manner and suggests the basic techniques for analyzing the algorithm. Figure 3 is the signal flow graph of the gradient algorithm.

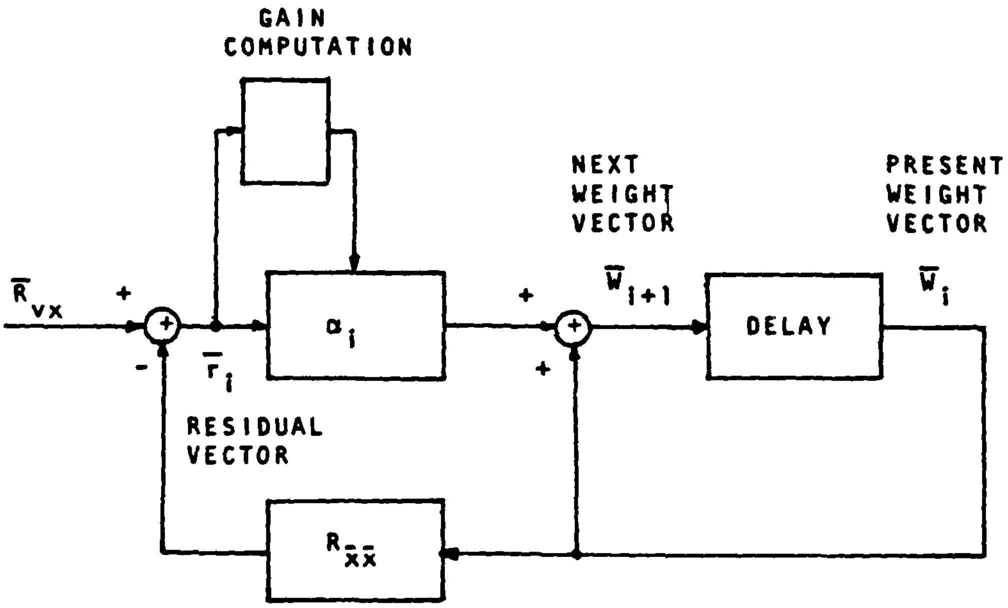


FIGURE 3. SIGNAL FLOW GRAPH OF GRADIENT DESCENT ALGORITHM

The gain computation box in Figure 3 is required for the steepest descent algorithm but is omitted for the simple gradient descent. Note that if the gain changes at each iteration the algorithm is time varying and becomes more difficult to analyze.

A practical problem with the steepest descent algorithm is that sample statistics must be formed to estimate $\overline{R}_{vx}$ and $R_{\overline{x}\overline{x}}$. The computational burden to obtain these estimates may be significant, and if the statistics are slowly varying they have to be periodically recomputed. Also, the sample statistics are only estimates and as such have a non zero variance.

We now address an extension of the gradient descent algorithm known as the Least Mean Square (LMS) adaptive algorithm. This algorithm constructs an estimate to the solution of the Normal Equation without constructing the sample covariance matrix.

V. LEAST MEAN SQUARE ALGORITHM:

The gradient descent technique described in the previous section requires the computation of the gradient of the mean square error function at each iteration. The LMS algorithm uses a noisy estimate to the gradient for its descent. The estimate is obtained by computing the gradient of the instantaneous error (rather than the gradient of the mean squared error). The squared error is defined in Eq (7b) and the first variation with respect to the weight vector $\overline{W}(k)$ is shown in Eq (19).

$$\begin{aligned}\delta e^2(k) &= -2\delta\overline{W}^T(k)\overline{X}(k)v(k) \\ &\quad + \delta\overline{W}^T(k)\overline{X}(k)\overline{X}^T(k)\overline{W}(k) \\ &\quad + \overline{W}^T(k)\overline{X}(k)\overline{X}^T(k)\delta\overline{W}(k) \qquad (19a)\\ &= -2\delta\overline{W}^T(k)[\overline{X}(k)v(k) - \overline{X}(k)\overline{X}^T(k)\overline{W}(k)] \qquad (19b)\\ &= -\delta\overline{W}^T(k)\cdot 2\overline{X}(k)[v(k) - \overline{X}^T(k)\overline{W}(k)] \qquad (19c)\end{aligned}$$

The gradient of the instantaneous error is shown in Eq (20).

$$\frac{\delta e^2(k)}{\delta\overline{W}^T(k)} = \nabla_{\overline{W}}[e^2(k)] = -2\overline{X}(k)[v(k) - \overline{X}^T(k)\overline{W}(k)] \qquad (20)$$

The estimated gradient is unbiased as demonstrated by forming its expected value. This is shown in Eq (21) which incorporates the definitions presented in Eqs (9) and (10).

$$E\{\nabla[e^2(k)]\} = -2E\{\overline{X}(k)[v(k) - \overline{X}^T(k)\overline{W}(k)]\} \tag{21a}$$

$$= -2[\overline{R}_{vx}(k) - R_{\bar{X}\bar{X}}\overline{W}(k)] \tag{21b}$$

Comparing equations (13) and (21b), we see

$$E\{\nabla[e^2(k)]\} = \nabla E\{e^2(k)\} \tag{21c}$$

The steepest descent LMS algorithm is essentially the same as the steepest descent gradient algorithm shown in Eq (18). In fact if we substitute the simple crossproduct $\overline{X}^T(k)\overline{X}(k)$ for the covariance matrix $E\{\overline{X}^T(k)\overline{X}(k)\}$ we obtain the steepest descent LMS algorithm as shown in Eq (22).

$$e(k) = v(k) - \overline{X}^T(k)\overline{W}(k) \tag{22a}$$

$$\overline{r}(k) = \overline{X}(k)e(k) \tag{22b}$$

$$\alpha(k) = \frac{1}{\overline{X}^T(k)\overline{X}(k)} \tag{22c}$$

$$\overline{W}(k+1) = \overline{W}(k) + \alpha(k)\overline{r}(k) \tag{22d}$$

Or more compactly;

$$\overline{W}(k+1) = \overline{W}(k) + \alpha(k)\overline{X}(k)[v(k) - \overline{X}^T(k)\overline{W}(k)] \tag{22e}$$

Note that equation (22) iterates on the time index "k" while equation (18) iterates on an auxiliary index "i". The LMS algorithm presented by Widrow employs a constant value of α and as such is a simple gradient descent as opposed to a steepest descent algorithm. The LMS steepest gradient algorithm can also be represented by a signal flow graph. This is presented in Figure 4.

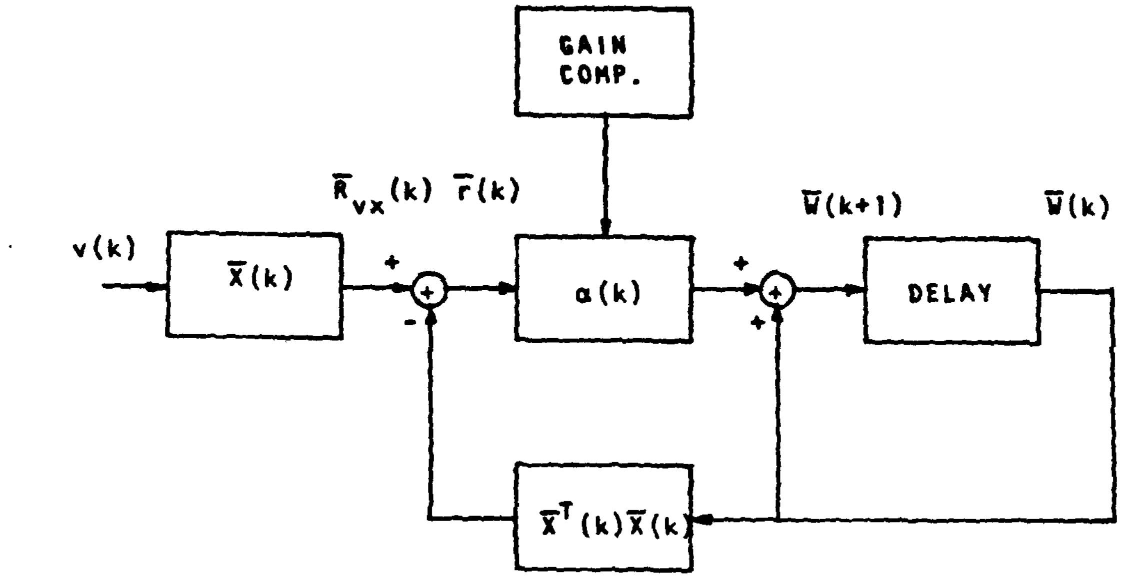


FIGURE 4a.

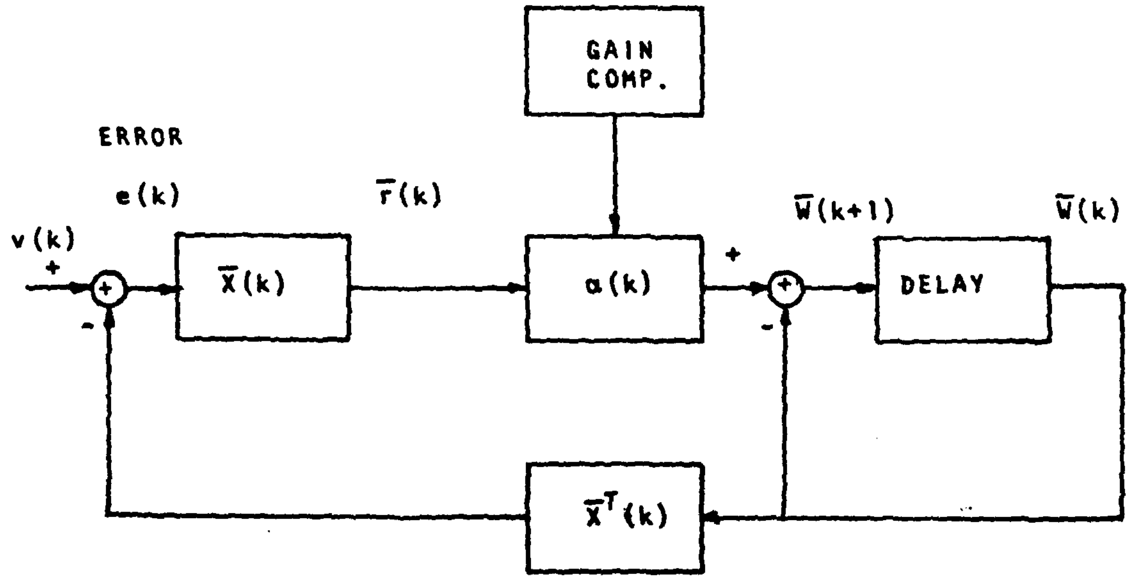


FIGURE 4b.

FIGURE 4a. SIGNAL FLOW GRAPH OF LMS DESCENT ALGORITHM
FIGURE 4b. FACTORED FORM OF SIGNAL FLOW GRAPH

Comparing Figures 4a and 3 we note that the difference between the algorithms lies in the use of approximations to construct the gradient (or residual). Since the approximations in Figure 4 are noisy estimates of the parameters of Figure 3, it stands to reason that the resultant gradients are also noisy estimates. Thus the LMS algorithm differs from the gradient algorithms by the presence of gradient noise within the outer feedback loop. see Figure 5.

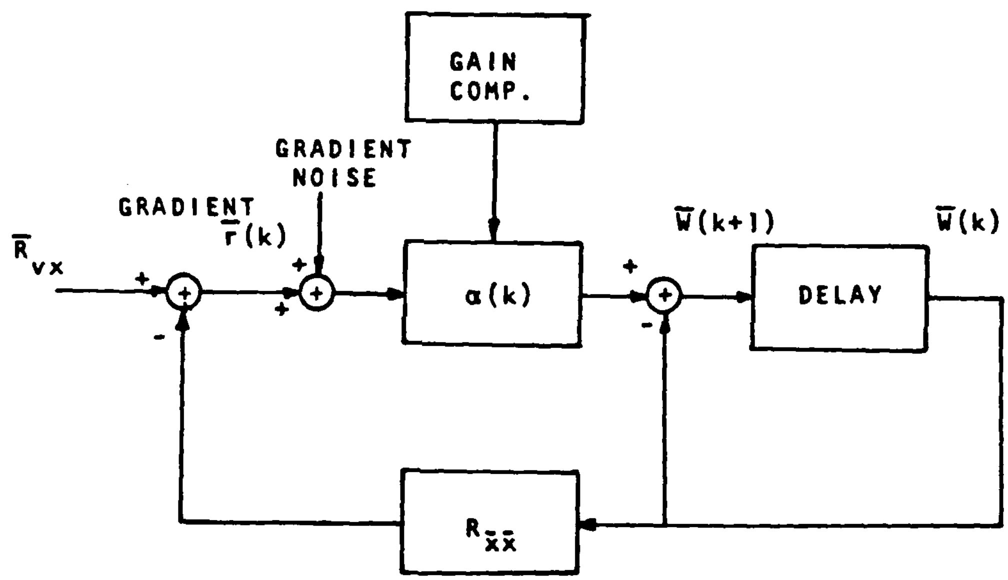


FIGURE 5. MODEL OF LMS ALGORITHM:
GRADIENT ALGORITHM WITH ADDITIONAL GRADIENT NOISE

The effect of the gradient noise on the steady state solution for the weight vector is the addition of an error component to the weight vector called misadjustment. Since the gradient noise is within the feedback loop, its effect on steady state response (misadjustment of weight vector or excess mean square error) will be scaled by the reciprocal of the loop gain. Hence for small steady state error, the loop gain should be large. This is accomplished by keeping the eigenvalues of the algorithm close to the unit circle. This in turn, is accomplished with small values

of the filter gain $\alpha(k)$. Small values of $\alpha(k)$ however, increase the transient time of the algorithm making it sluggish and non-responsive to changes in data statistics. We now continue with another formulation of the adaptive filter for which internal gain adjustments allow for low values of misadjustment without excessively long adaptation times.

VI. OPTIMAL LINEAR COMBINER:

We now address the following question; Given an apriori optimal estimate to the weight vector $\overline{W}(k)$, and the desired output $v(k)$, what is the optimum linear combination of this information to compute the next estimate of the weight vector? An equivalent question is, determine the form of the linear gains L and G as shown in equation (23).

$$\overline{W}(k+1) = L\,\overline{W}(k) + \overline{G}\,v(k) \tag{23}$$

Where $v(k)$ satisfies Eq (6), repeated here as Eq (24).

$$v(k) = \overline{W}^T(k)\overline{X}(k) + e(k) \tag{24}$$

First let us substitute Eq (24) into Eq (23) and then examine the expected value of $\overline{W}(k+1)$.

$$E\{\overline{W}(k+1)\} = E\{L\,\overline{W}(k) + \overline{G}\,v(k)\} \tag{25a}$$

$$= E\{L\,\overline{W}(k) + \overline{G}\,\overline{X}^T(k)\overline{W}(k) + \overline{G}\,e(k)\} \tag{25b}$$

$$= [L + \overline{G}\,\overline{X}^T(k)]\,E\{\overline{W}(k)\} + \overline{G}\,E\{e(k)\} \tag{25c}$$

By assumption; $E\{e(k)\} = 0$, and $E\{W(k+1)\} = E\{W(k)\} = W$. We then have;

$$L + \overline{G}\,\overline{X}^T(k) = I \tag{26a}$$

or

$$L = I - \overline{G}\,\overline{X}^T(k) \tag{26b}$$

Substituting Eq (26) into Eq (23), we obtain Eq (27).

$$\overline{W}(k+1) = L\ \overline{W}(k) + \overline{G}\ v(k) \tag{27a}$$

$$= [I - \overline{G}\ \overline{X}^T(k)]\overline{W}(k) + \overline{G}\ v(k) \tag{27b}$$

$$\text{Or } \overline{W}(k+1) = \overline{W}(k) + \overline{G}[v(k) - \overline{X}^T(k)\overline{W}(k)] \tag{27c}$$

Comparing Eq (27c) with Eq (22), we find the two forms remarkably similar. It now remains to derive the gain term $\overline{G}$.

Recalling the concept of misadjustment, we recognize that any error $e(k)$ computed during an iteration can come from two sources. The first is simply the error due to misadjustment of the weights. In that case, the weights should be adjusted. The second source of error is the noise component in the data vector $\overline{X}(k)$. This noise propogates to the filter output even if the weights are set to the optimal values. For that source of noise, we should not adjust the weight vector. We now consider the cost function of Eq (28) which is structured to reflect the two sources of uncertainty leading to the filter error.

$$J = [v(k) - \overline{X}^T(k)\overline{W}(k+1)]^2 + [\overline{W}(k+1) - \overline{W}(k)]^T B^{-1}(k)\ [\overline{W}(k+1) - \overline{W}(k)] \tag{28}$$

The matrix $B(k)$ is yet to be defined, but will reflect our desire to penalize (or emphasize) changes in the weight vector relative to the prediction error. $B(k)$ will be termed the information matrix as it will reflect our apriori information as to the likely source of instantaneous errors.

Now let us minimize the cost function J with respect to the weight vector $\overline{W}(k+1)$. In Eq (29), we take the first variation of J with respect to $\overline{W}(k+1)$.

$$\delta J = -2\delta\overline{W}^T(k+1)\overline{X}(k)[v(k) - \overline{X}^T(k)\overline{W}(k+1)] +$$

$$2\delta\overline{W}^T(k+1)\ B^{-1}[\overline{W}(k+1) - \overline{W}(k)] \tag{29a}$$

$$= 2\delta\overline{W}^T(k+1)\{[B^{-1}(k) + \overline{X}(k)\overline{X}^T(k)]\overline{W}(k+1)$$

$$- [B^{-1}(k)\overline{W}(k) + \overline{X}(k)v(k)]\} \tag{29b}$$

The first variation is zero if Eq (30) holds true.

$$[B^{-1}(k) + \overline{X}(k)\overline{X}^T(k)]\overline{W}(k+1) = [B^{-1}(k)\overline{W}(k) - \overline{X}(k)v(k)] \tag{30a}$$

Or $\overline{W}(k+1) = [B^{-1}(k) + \overline{X}(k)\overline{X}^T(k)]^{-1}[B^{-1}(k)\overline{W}(k) - \overline{X}(k)v(k)]$ (30b)

Now apply the inversion Lemma shown in Eq (31)

$$(\Lambda_1 + \Lambda_{12}\Lambda_2{}^{-1}\Lambda_{21})^{-1} =$$

$$\Lambda_1{}^{-1} - \Lambda_1{}^{-1}\Lambda_{12}[\Lambda_2 + \Lambda_{21}\Lambda_1{}^{-1}\Lambda_{12}]^{-1}\Lambda_{21}\Lambda_1 \tag{31}$$

with the substitutions indicated in Eq (32), we obtain Eq (33).

$$\Lambda_1 = B^{-1}(k) \tag{32a}$$

$$\Lambda_{12} = \overline{X}(k) \tag{32b}$$

$$\Lambda_{21} = \overline{X}^T(k) \tag{32c}$$

$$\Lambda_2{}^{-1} = 1 \tag{32d}$$

$$\overline{W}(k+1) = \{B(k) - B(k)\overline{X}(k)[1 + \overline{X}^T(k)B(k)\overline{X}(k)]^{-1}\overline{X}^T(k)\overline{X}(k)\}\cdot$$

$$[B^{-1}(k)\overline{W}(k) + \overline{X}(k)v(k)] \tag{33a}$$

Expanding and gathering terms, we obtain Eq (33b).

$$\overline{W}(k+1) = \overline{W}(k) + B(k)\overline{X}(k)v(k)$$

$$- \frac{B(k)\overline{X}(k)\overline{X}^T(k)\overline{W}(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)}$$

$$- \frac{B(k)\overline{X}(k)\overline{X}^T(k)B(k)\overline{X}(k)v(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} \quad (33b)$$

$$= \overline{W}(k) + \frac{B(k)\overline{X}(K)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} [v(k) - \overline{X}^T(k)\overline{W}(k)] \quad (33c)$$

Or $$\overline{W}(k+1) = \overline{W}(k) + \frac{B(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} \overline{X}(k)[v(k)-\overline{X}^T(k)\overline{W}(k)] \quad (33d)$$

Comparing Eq (33d) with Eq (27), we have determined the gain term of Eq (27) to be;

$$G = \frac{B(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} \overline{X}(k) \quad (34)$$

and comparing Eq (33) with Eq (22), we recognize that the convergence factor $\alpha(k)$ which determines the size of the step in the gradient direction has been replaced with the scaled matrix indicated in Eq (35).

$$\alpha(k) \Rightarrow \frac{1}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} B(k) \quad (35)$$

Note that the weight update scheme presented in Eq (33) constructs the sample gradient $\overline{X}(k)[v(k)-\overline{X}^T(k)\overline{W}(k)]$ as did the LMS scheme but then subjects the gradient to the matrix operator indicated in Eq (35). The particular form that the operator takes is dependent upon the selection of the information matrix B(k) (or $B^{-1}(k)$). The operator allows for the option of rotating the gradient vector. This class of algorithm is called a gradient deflection scheme. The extent of the deflection is of course controlled by the information matrix B(k).

VII SELECTION OF THE INFORMATION MATRIX B(k):

VII.A Minimum Variance Filter.

The first approach we examine will be to determine the B(k) which minimizes the variance of the estimate $\overline{W}(k)$. Thus we need to minimize Eq (36), (where $\overline{W}^*$ is the optimal value

$$E\{[\overline{W}(k+1)-\overline{W}^*][\overline{W}(k+1)-\overline{W}^*]^T\} \doteq S_{\overline{W}\overline{W}}(k+1) \tag{36}$$

of the weight vector) subject to the update algorithm shown in Eq (37).

$$\overline{W}(k+1) = \overline{W}(k) + G(k)[v(k)-\overline{X}^T(k)\overline{W}(k)] \tag{37}$$

The definition of the error term corresponding to the optimal weight vector $\overline{W}^*$ is shown in Eq (38).

$$v(k) = \overline{X}^T(k)\overline{W}^* + e(k) \tag{38}$$

Then the error vector $[\overline{W}(k+1) - \overline{W}^*]$ is shown in Eq (39).

$$[\overline{W}(k+1)-\overline{W}^*] = [\overline{W}(k)-\overline{W}^*] - G(k)\overline{X}^T(k)[\overline{W}(k)-\overline{W}^*] + G(k)e(k) \tag{39a}$$

$$= [I - G(k)\overline{X}^T(k)][\overline{W}(k)-\overline{W}^*] + G(k)e(k) \tag{39b}$$

The variance can now be found by substitution of Eq (39b) into Eq (36).

$$S_{\overline{W}\overline{W}}(k+1) = E\{[\overline{W}(k+1)-\overline{W}^*][\overline{W}(k+1)-\overline{W}^*]^T\} \tag{40a}$$

$$= E\Big\{\{[I-G(k)\overline{X}^T(k)][\overline{W}(k)-\overline{W}^*]+G(k)e(k)\}\cdot \{[I-G(k)\overline{X}^T(k)][\overline{W}(k)-\overline{W}^*]+G(k)e(k)\}^T\Big\} \tag{40a}$$

$$= E\{[I-G(k)\bar{X}^T(k)][\bar{W}(k)-\bar{W}^*][\bar{W}(k)-\bar{W}^*]^T[I-G(k)\bar{X}^T(k)]^T\}$$

$$+ E\{[I-G(k)\bar{X}^T(k)][\bar{W}(k)-\bar{W}^*]G(k)e(k)\}$$

$$+ E\{e(k)G(k)\{[\bar{I}-G(k)\bar{X}^T(k)][\bar{W}(k)-\bar{W}^*]\}^T\}$$

$$+ E\{G(k)G^T(k)e^2(k)\} \tag{40b}$$

Moving the expected value operator through the right hand side of Eq (40b) will lead to Eq (41), where $E\{e^2(k)\} = R_{ee}(k)$.

$$S_{\bar{w}\bar{w}}(k+1) = [I-G(k)\bar{X}^T(k)]S_{\bar{w}\bar{w}}(k)[I-G(k)\bar{X}^T(k)]$$

$$+ G(k)G^T(k)R_{ee}(k) \tag{41}$$

Expanding Eq (41), we obtain the resits shown in Eq (42).

$$S_{\bar{w}\bar{w}}(k+1) = S_{\bar{w}\bar{w}}(k)+G(k)\bar{X}^T(k)S_{\bar{w}\bar{w}}(k)\bar{X}(k)G^T(k)$$

$$+G(k)G^T(k)R_{ee}(k)$$

$$-G(k)\bar{X}^T(k)S_{\bar{w}\bar{w}}(k)$$

$$-S_{\bar{w}\bar{w}}(k)\bar{X}(k)G^T(k) \tag{42a}$$

$$= S_{\bar{w}\bar{w}}(k)+G(k)[\bar{X}^T(k)S_{\bar{w}\bar{w}}(k)\bar{X}(k)+R_{ee}(k)]G^T(k)$$

$$-G(k)\bar{X}^T(k)S_{\bar{w}\bar{w}}(k)-S_{\bar{w}\bar{w}}(k)\bar{X}(k)G^T(k) \tag{42b}$$

For the purpose of further manipulation of this expression, we define the scalar shown in Eq (43).

$$d(k) = [\bar{X}^T(k)S_{\bar{w}\bar{w}}(k)\bar{X}(k)+R_{ee}(k)] \tag{43}$$

Now rewrite Eq (42) in the form indicated as Eq (44).

$$S_{\bar{w}\bar{w}}(k+1) = G(k)d(k)G^T(k) + S_{\bar{w}\bar{w}}(k)$$
$$- G(k)d(k)d^{-1}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)$$
$$- S_{\bar{w}\bar{w}}(k)\overline{X}(k)d^{-1}(k)d(k)G^T(k)$$
$$+ S_{\bar{w}\bar{w}}(k)\overline{X}(k)d^{-1}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)$$
$$- S_{\bar{w}\bar{w}}(k)\overline{X}(k)d^{-1}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k) \quad (44a)$$

$$S_{\bar{w}\bar{w}}(k+1) = [G(k)-S_{\bar{w}\bar{w}}(k)\overline{X}(k)d^{-1}(k)]d(k)[G^T(k)-d^{-1}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)]$$
$$+ S_{\bar{w}\bar{w}}(k)[I-\overline{X}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)] \quad (44b)$$

Now define a new cost function J(G) by Eq (45)

$$J(G) = TR[S_{\bar{w}\bar{w}}(k+1)] \quad (45a)$$
$$= TR\{[G-S_{\bar{w}\bar{w}}\overline{X}d^{-1}]d[G^T-d^{-1}\overline{X}^TS_{\bar{w}\bar{w}}] + S_{\bar{w}\bar{w}}[I-\overline{X}\overline{X}^TS_{\bar{w}\bar{w}}]\} \quad (45b)$$

We have temporarily suppressed the indces of Eq (45) for the economy of notation. TR( ) indicates the trace of the associated matrix. We now form the first variation of the Trace with respect to the gain matrix G.

$$\delta J(G) = TR\{\delta[G-S_{\bar{w}\bar{w}}\overline{X}d^{-1}]d[G^T-d^{-1}\overline{X}^TS_{\bar{w}\bar{w}}] +$$
$$[G-S_{\bar{w}\bar{w}}\overline{X}d^{-1}]d\delta[G^T-d^{-1}\overline{X}^TS_{\bar{w}\bar{w}}]\} \quad (46a)$$
$$= TR\{\delta Gd[G^T-d^{-1}\overline{X}^TS_{\bar{w}\bar{w}}] + [G-S_{\bar{w}\bar{w}}\overline{X}d^{-1}]d\delta G\} \quad (46b)$$
$$= 2TR\{\delta Gd[G^T-d^{-1}\overline{X}^TS_{\bar{w}\bar{w}}]\} \quad (46c)$$

The first variation of Eq (46c) will be zero if Eq (47) holds.

$$d(k)[G^T(k)-d^{-1}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}] = 0 \quad (47a)$$

$$d(k)G^T(k) - \overline{X}^T(k)S_{\bar{w}\bar{w}} = 0 \tag{47b}$$

Or
$$G(k) = d^{-1}(k)S_{\bar{w}\bar{w}}\overline{X}(k) \tag{47c}$$

Now substituting Eq (43) back into Eq (47c), we have the gain term G(k) which corresponds to the minimum variance filter.

$$G(k) = \frac{S_{\bar{w}\bar{w}}}{R_{ee}(k) + \overline{X}^T(k)S_{\bar{w}\bar{w}}\overline{X}(k)}\overline{X}(k) \tag{48a}$$

$$= \frac{\frac{1}{R_{ee}} S_{\bar{w}\bar{w}}}{1 + \overline{X}(k)\frac{1}{R_{ee}} S_{\bar{w}\bar{w}} \overline{X}(k)}\overline{X}(k) \tag{48b}$$

Comparing Eq (48b) with Eq (34), we recognize that the information matrix B(k) which realizes the minimum variance weights while minimizing the cost function of Eq (27) is simply the weight covariance matrix scaled by the reciprocal of the error power of the filter. This is indicated in Eq (49).

$$B(k) = \frac{1}{R_{ee(k)}} S_{\bar{w}\bar{w}}(k+1) \tag{49}$$

We must now construct the weight covariance matrix $S_{\bar{w}\bar{w}}(k+1)$. This is accomplished by substituting the minimum variance information matrix G(k) back into Eq (41). This is done in Eq (50).

$$S_{\bar{w}\bar{w}}(k+1) = S_{\bar{w}\bar{w}}(k) + G(k)d(k)G^T(k) - G(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k) - S_{\bar{w}\bar{w}}(k)\overline{X}(k)G^T(k) \tag{50a}$$

$$S_{\bar{w}\bar{w}}(k+1) = S_{\bar{w}\bar{w}}(k) + d^{-1}(k)S_{\bar{w}\bar{w}}(k)\overline{X}(k)d(k)d^{-1}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)$$
$$- d^{-1}(k)S_{\bar{w}\bar{w}}(k)\overline{X}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)$$
$$- d^{-1}(k)S_{\bar{w}\bar{w}}(k)\overline{X}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k) \qquad (50b)$$

$$S_{\bar{w}\bar{w}}(k+1) = S_{\bar{w}\bar{w}}(k) - \frac{S_{\bar{w}\bar{w}}(k)\overline{X}(k)\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)}{R_{ee}(k)+\overline{X}^T(k)S_{\bar{w}\bar{w}}(k)\overline{X}(k)} \qquad (50c)$$

If we rewrite Eq (50c) as Eq (50d) and again apply the inversion lemma with the substitutions indicated in Eq (50e) we will have the expression shown in Eq (51).

$$S_{\bar{w}\bar{w}}(k+1) = S_{\bar{w}\bar{w}}(k) - S_{\bar{w}\bar{w}}(k)\overline{X}(k)[R_{ee}(k)+X^T(k)S_{\bar{w}\bar{w}}(k)\overline{X}(k)]^{-1}X^T(k)S_{ww}(k)$$
$$(50d)$$

$$\begin{aligned} \Lambda_1^{-1} &= S_{\bar{w}\bar{w}}(k) \\ \Lambda_{12} &= \overline{X}(k) \\ \Lambda_{21} &= \overline{X}^T(k) \\ \Lambda_2 &= R_{ee}(k) \end{aligned} \qquad (50e)$$

$$[\Lambda_1 + \Lambda_{12}\Lambda_2^{-1}\Lambda_{21}]^{-1} = \Lambda_1^{-1} - \Lambda_1^{-1}\Lambda_{12}[\Lambda_2 + \Lambda_{21}\Lambda_1^{-1}\Lambda_{12}]^{-1}\Lambda_{21}\Lambda_1^{-1}$$

$$S_{\bar{w}\bar{w}}(k+1) = [S_{\bar{w}\bar{w}}^{-1}(k) + \frac{1}{R_{ee}(k)}\overline{X}(k)\overline{X}^T(k)]^{-1} \qquad (51a)$$

Or $$S_{\bar{w}\bar{w}}^{-1}(k+1) = S_{\bar{w}\bar{w}}^{-1}(k) + \frac{1}{R_{ee}(k)}\overline{X}(k)\overline{X}^T(k) \qquad (51b)$$

Equation (51) is not as convenient for computational work as is Eq (50). Equation (51) does however offer insight into the behavior of the covariance matrix $S_{\bar{w}\bar{w}}(k+1)$ as the algorithm iterates. A matrix signal flow graph corresponding to Eq (51) is shown in Figure 6.

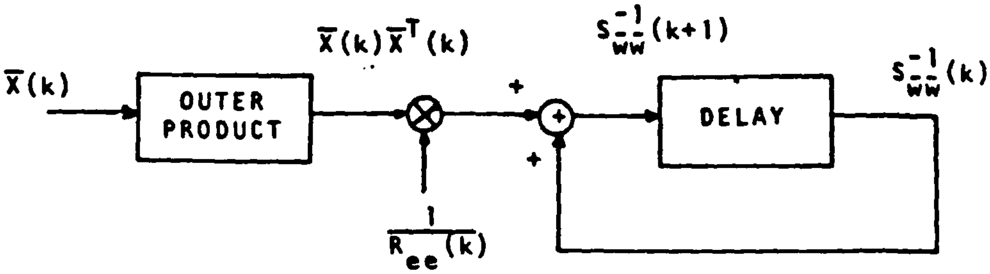


FIGURE 6. SIGNAL FLOW GRAPH INVERSE COVARIANCE MATRIX UPDATE ALGORITHM [EQ(51)]

We can surmize the behavior of Eq (51) by examining the structure of Figure 6. We see that the inverse covariance matrix $S^{-1}_{\overline{w}\overline{w}}(k+1)$ is simply the output of a matrix integrator. As the iteration progresses, the diagonal terms of the matrix increases without bound, hence the inverse matrix $S_{\overline{w}\overline{w}}(k+1)$ approaches zero. We cantrace this effect back to the cost function of Eq (29) from which we constructed this algorithm. We see that as the inverse covariance matrix becomes arbitrarily large with large time index, the cost function penalizes changes in the weight vector. In a sense, because the covariance matrix $S_{\overline{w}\overline{w}}(k)$ is approaching zero, the algorithm is convinced that the weights are correct and should not be changed. The filter reflects this position by driving the feedfoward gain [Eq (48)] to zero effectively preventing changes in the weight vector.

VII.B <u>Fading Memory Filter</u>.

In the previous section we selected the information matrix B(k) to be equal to the scaled version of the weight covariance matrix. We saw that the covariance matrix was constructed iteratively by a matrix integrator as shown in Figure 6 and described below in Eq (52).

$$B(k+1) = S_{\bar{w}\bar{w}}(k+1) = \left[\sum_{i=0}^{k} \frac{\bar{X}(i)\bar{X}^T(i)}{R_{ee}(i)}\right]^{-1} \tag{52a}$$

$$= \left[S^{-1}_{\bar{w}\bar{w}}(k) + \frac{X(k)X^T(k)}{R_{ee}(k)}\right]^{-1} \tag{52b}$$

We alluded to a potential problem with this selection of the information matrix. Namely, that the filter turns itself off as the information matrix goes towards zero. We now consider a varaiation of the information matrix which prevents the filter from shutting down. We take a hint from Eq (52) and Figure 6 and define an exponentially weighted summation in Eq (53).

$$B(k+1) = P_{\bar{w}\bar{w}}(k+1) = \left[\sum_{i=0}^{k} \frac{\bar{X}(i)\bar{X}^T(i)}{R_{ee}(i)} \; q^{(k-i)}\right]^{-1} \tag{53a}$$

$$= \left[\sum_{i=0}^{k-1} \frac{\bar{X}(i)\bar{X}^T(i)}{R_{ee}(i)} \; q^{(k-i)} + \frac{\bar{X}(k)\bar{X}^T(k)}{R_{ee}(k)}\right]^{-1} \tag{53b}$$

$$= \left[q \sum_{i=0}^{k-1} \frac{X(i)X^T(i)}{R_{ee}(i)} \; q^{(k-1-i)} + \frac{X(k)X^T(k)}{R_{ee}(k)}\right]^{-1} \tag{53c}$$

Or

$$P_{\bar{w}\bar{w}}(k+1) = \left[q \, P^{-1}_{\bar{w}\bar{w}}(k) + \frac{\bar{X}(k)\bar{X}^T(k)}{R_{ee}(k)}\right]^{-1} \tag{53d}$$

We recognize that Eq (53d) describes an integrator with other than unity feedback. If the scalar "q" is less than unity, the integrator is called a leaky integrator. A signal flow graph of Eq (53d) is shown in Figure 7.

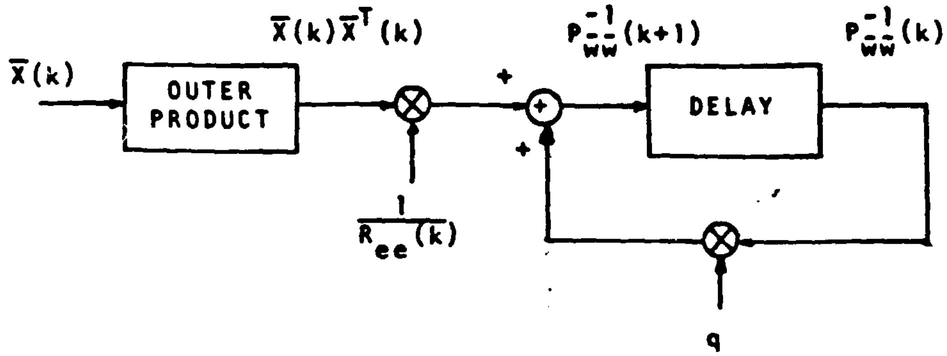


FIGURE 7. SIGNAL FLOW GRAPH OF LEAKY INTEGRATOR REPRESENTATION OF INFORMATION MATRIX

Note that the leaky integrator's output does not increase without bound as does the unity gain integrator but is limited to a steady state gain of $(1-q)^{-1}$. Thus the gain of the algorithm does not approach zero, but rather to small values proportional to (1-q). We refer to the parameter "q" as the fade factor of the algorithm.

If we apply the inversion lemma to Eq (53) with the following substitutions we obtain the equivalent form of the information matrix update as indicated in Eq (54).

$$\Lambda_1 = q\, P^{-1}_{\underline{w}\underline{w}}(k) \tag{54a}$$

$$\Lambda_{12} = \overline{X}(k) \tag{54b}$$

$$\Lambda_{21} = \overline{X}^T(k) \tag{54c}$$

$$\Lambda_2 = \frac{1}{R_{ee}(k)} \tag{54d}$$

$$[\Lambda_1+\Lambda_{12}\Lambda_2^{-1}\Lambda_{21}]^{-1} = \Lambda_1^{-1}-\Lambda_1^{-1}\Lambda_{12}[\Lambda_2+\Lambda_{21}\Lambda_1^{-1}\Lambda_{12}]\Lambda_{21}\Lambda_1^{-1} \tag{54e}$$

$$P_{\underline{w}\underline{w}}(k+1) = \frac{1}{q}\, P_{\underline{w}\underline{w}}(k) + \frac{q^{-1}P_{\underline{w}\underline{w}}(k)\overline{X}(k)\overline{X}^T(k)q^{-1}P_{\underline{w}\underline{w}}(k)}{R_{ee}(k) + \overline{X}^T(k)q^{-1}P_{\underline{w}\underline{w}}(k)\overline{X}(k)} \tag{54f}$$

Or $$P_{\bar{w}\bar{w}}(k+1) = \frac{1}{q} \quad P_{\bar{w}\bar{w}}(k) - \frac{P_{\bar{w}\bar{w}}(k)\bar{X}(k)\bar{X}^T(k)P_{\bar{w}\bar{w}}(k)}{qR_{ee}(k)+\bar{X}^T(k)P_{\bar{w}\bar{w}}(k)\bar{X}(k)} \qquad (54g)$$

Comparing Eq (54g) to Eq (50) and Figure 7 to Figure 6, we of course recognize that as the parameter "q" goes to unity, the leaky information matrix becomes the minimum variance information matrix.

VII.C Steepest Descent Filter.

Now suppose we select the information matrix $B(k)$ to be K times the identity matrix [K I] where K is a large scalar. Thus $B^{-1}(k) = \frac{1}{K} I$. This matrix has the effect in the penalty function Eq (27) of not penalizing changes in the weight vector. This is equivalent to allowing the weight vector to make large changes at each iteration. Substituting this B(k) into Eq (33c) we obtain Eq (55).

$$\bar{W}(k+1) = \bar{W}(k) + \frac{K\ I}{1+\bar{X}^T(k)KI\ \bar{X}(k)} \bar{X}(k)[v(k)-\bar{X}^T(k)\bar{W}(k)] \quad (55a)$$

$$= \bar{W}(k) + \frac{1}{K^{-1} + \bar{X}^T(k)\bar{X}(k)} \bar{X}(k)[v(k)-\bar{X}^T(k)\bar{W}(k)] \quad (55b)$$

We note that as K becomes large so that $\frac{1}{K}$ becomes small with respect to the inner product $\bar{X}^T(k)\bar{X}(k)$, Eq (55b) approaches Eq (56).

$$\bar{W}(k+1) = \bar{W}(k) + \frac{1}{\bar{X}^T(k)\bar{X}(k)} \bar{X}(k)[v(k)-\bar{X}^T(k)\bar{W}(k)] \qquad (56)$$

We recognize Eq (56) is equivalent to the steepest descent LMS algorithm of Eq (22). Note that this filter results from the penalty function interpreting the mean square error as arising from errors in the weight vector. This is equivalent

to interpreting the scalar K as $R_{ee}^{-1}(k)$ the power in the prediction error of the optimal vector weights. Then the information matrix $B(k) = K\ I$ becomes $R_{ee}^{-1}(k)\ I$. If K is large, the interpretation is that the optimal prediction error is small hence the filter errors are indeed due to an incorrect weight vector.

VII.D. <u>LMS Filter</u>.

Now suppose we select the information matrix B(k) to be μ times the identity matrix [μ I] where μ is a small scalar. Thus $B^{-1}(k) = \frac{1}{\mu}\ I$. This has the effect in the penalty function Eq (27) of severely penalizing changes in the weight vector or equivalently of preventing the weight vector from having large changes at each iteration. Substituting this B(k) into Eq (33c) we obtain Eq (57).

$$\overline{W}(k+1) = \overline{W}(k) + \frac{\mu\ I}{1 + \overline{X}^T(k)\mu I\ \overline{X}(k)}\ \overline{X}(k)[v(k)-\overline{X}^T(k)\overline{W}(k)] \quad (57a)$$

$$= \overline{W}(k) + \frac{\mu}{1 + \mu\overline{X}^T(k)\overline{X}(k)}\ \overline{X}(k)[v(k)-\overline{X}^T(k)\overline{W}(k)] \quad (57b)$$

We note that as μ becomes sufficiently small to force the product $\mu\overline{X}^T(k)\overline{X}(k)$ to be small compared to unity, Eq (57) approaches Eq (58).

$$\overline{W}(k+1) = \overline{W}(k) + \mu\overline{X}(k)[v(k)-\overline{X}^T(k)\overline{W}(k)] \quad (58)$$

We recognize Eq (58) is equivalent to Widrow's LMS algorithm. Note that this filter results from the penalty function interpreting the mean square error as arizing from prediction noise. This is equivalent to interpreting the scalar μ as $R_{ee}^{-1}(k)$ the power in the prediction error for the optimal vector weights. Then the information matrix $B(k) = \mu\ I$ becomes $R_{ee}^{-1}(k)\ I$. If the scalar μ is small the interpretation is that the optimal weight prediction error is large. That is very noisy data! Thus changes in the weight vector will have little effect in reducing the error.

To reiterate this section, we have found the optimal linear combiner to be of the form shown in Figure 8. and in Eq (59).

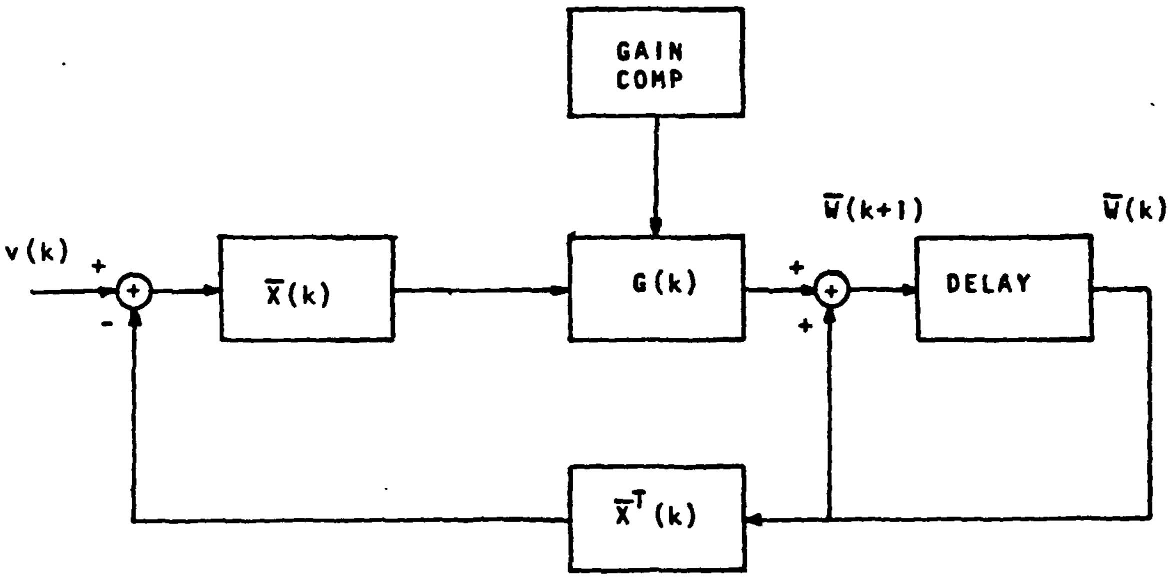


FIGURE 8. SIGNAL FLOW GRAPH OF OPTIMAL LINEAR COMBINER

$$\bar{W}(k+1) = \bar{W}(k) + G(k)\ \bar{X}(k)[v(k)-\bar{X}^T(k)\bar{W}(k)] \tag{59a}$$

Where
$$G(k) = \frac{B(k)}{1 + \bar{X}^T(k)B(k)\bar{X}(k)} \tag{59b}$$

The B(k)'s of interset to us are listed below;

1. $B(k) = \frac{1}{R_{ee}(k)} S_{\bar{w}\bar{w}}(k)$ [Minimum Variance] (59c)

Where $S_{\bar{w}\bar{w}}(k+1) = [S^{-1}_{\bar{w}\bar{w}}(k) + \frac{1}{R_{ee}(k)} \bar{X}^T(k)\bar{X}(k)]^{-1}$ (59d)

2. $B(k) = \frac{1}{R_{ee}(k)} P_{\bar{w}\bar{w}}(k)$ [Fading Memory] (59e)

Where $P_{\bar{w}\bar{w}}(k+1) = [q\ P^{-1}_{\bar{w}\bar{w}}(k) + \frac{1}{R_{ee}(k)} \bar{X}^T(k)\bar{X}(k)]^{-1}$ (59f)

3. $B(k) = K\ I$, $K >> 1$ [= Steepest Descent] (59g)

4. $B(k) = \mu\ I$, $\mu << 1$ [= LMS] (59h)

IIX.. STEADY STATE BEHAVIOR OF OPTIMAL LINEAR COMBINER:

We will now examine the steady state behavior of the linear combiner. We start by a form of the algorithm for the weight vector update which emphasizes the weight error vector. The weight error vector is defined as the difference between the optimal weight vector $\overline{W}^*$ and the present weight vector $\overline{W}(k+1)$. We can rewrite Eq (59a) as Eq (60).

$$\overline{W}(k+1) = \overline{W}(k) + \Delta\overline{W}(k+1) \tag{60a}$$

Where

$$\Delta\overline{W}(k+1) = G(k)\ \overline{X}(k)[v(k)-\overline{X}^T(k)\overline{W}(k)] \tag{60b}$$

We define the weight error vector or the misalignment vector $\overline{h}(k+1)$ in Eq (61).

$$\overline{h}(k+1) = \overline{W}^* - \overline{W}(k+1) \tag{61}$$

We can manipulate Eq (60) and substitute Eq (61) to obtain the results shown in Eq (62).

$$\Delta\overline{W}(k+1) = \overline{W}(k+1) - \overline{W}(k) \tag{62a}$$

$$= [\overline{W}^*-\overline{h}(k+1)] - [\overline{W}^*-\overline{h}(k)] \tag{62b}$$

$$= -[\overline{h}(k+1) - \overline{h}(k)] \tag{62c}$$

$$= -\Delta\overline{h}(k+1) \tag{62d}$$

Thus the change in the weight vector is the negative of the change in the misalignment vector. This is shown in Eq 63).

$$\Delta\bar{h}(k+1) = -\Delta\bar{W}(k+1) \tag{63a}$$

$$= -G(k)\bar{X}(k)\{v(k)-\bar{X}^T(k)[\bar{W}^*-\bar{h}(k)]\} \tag{63b}$$

$$= G(k)\bar{X}(k)\{\bar{X}^T(k)\bar{W}^*-v(k)-\bar{X}^T(k)\bar{h}(k)\} \tag{63c}$$

We can substitute Eq (38) repeated here as Eq (64) into Eq (63c) and obtain Eq (65) which is the difference equation for the misalignment vector.

$$\bar{X}^T(k)\bar{W}^* - e(k) = v(k) \tag{64a}$$

Hence $\Delta\bar{h}(k+1) = G(k)\bar{X}(k)[e(k)-\bar{X}^T(k)\bar{h}(k)]$ (64b)

Thus $\bar{h}(k+1) = \bar{h}(k) + G(k)\bar{X}(k)[e(k)-\bar{X}^T(k)\bar{h}(k)]$ (65)

A signal flow graph representing Eq (65) is presented in Figure 9.

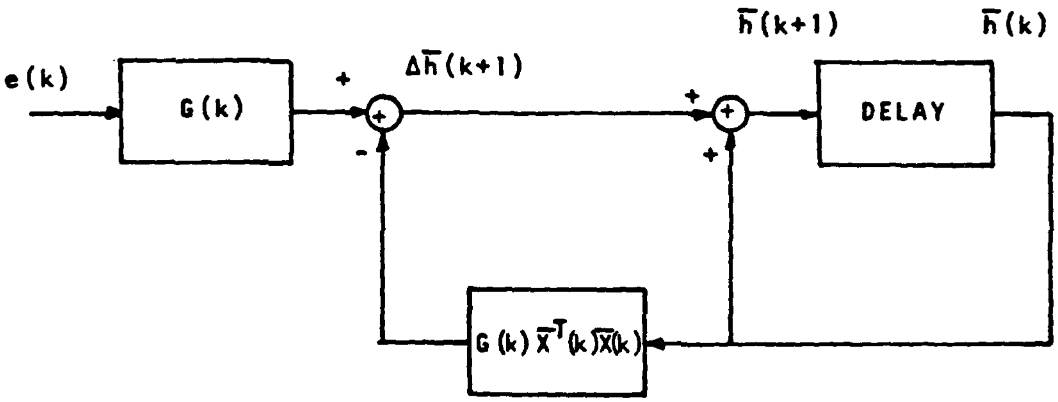


FIGURE 9. SIGNAL FLOW GRAPH FOR MISALIGNMENT VECTOR

Let us examine the steady state behavior of the misalignment vector. In steady state, the expected value of the change in misalignment is zero. See Eq (66).

$$E\{\Delta\bar{h}(k+1)\} = 0 \tag{66a}$$

$$E\{G(k)\bar{X}(k)[e(k)-\bar{X}^T(k)\bar{h}(k)]\} = 0 \tag{66b}$$

$$E\{G(k)\bar{X}(k)e(k)\} - E\{G(k)\bar{X}(k)\bar{X}^T(k)\bar{h}(k)\} = 0 \tag{66c}$$

From which we conclude the following;

$$E\{\overline{h}(k)\} = E\{G(k)\overline{X}(k)\overline{X}^T(k)\}^{-1}E\{G(k)\overline{X}(K)\}E\{e(k)\} \qquad (67a)$$

Or $$E\{\overline{h}(k)\} = \overline{0} \qquad (67b)$$

Thus we find that the expected misalignment is zero. This is true for any gain matrix providing of course the filter is stable. We have anticipated this result earlier when we demonstrated that $\overline{W}(k)$ is an unbiased estimate of $\overline{W}^*$, the optimal weight vector.

IV. CONVERGENCE PROPERTIES OF ALGORITHM.

Now let us consider the transformation of variables which diagonalizes the matrix $G(k)\overline{X}(k)\overline{X}^T(k)$. Let the new coordinate system be defined by Eq (68).

$$\overline{h}(k) = Q\overline{h}'(k) \qquad (68a)$$

Then $$Q\overline{h}'(k+1) = Q\overline{h}'(k)+G(k)\overline{X}(k)e(k)-G(k)\overline{X}(k)\overline{X}^T(k)Q\overline{h}'(k) \qquad (68a)$$

Or $$\overline{h}'(k+1) = \overline{h}'(k) + Q^{-1}G(k)\overline{X}(k)e(k)$$

$$- Q^{-1}G(k)\overline{X}(k)\overline{X}^T(k)Q\overline{h}'(k) \qquad (68c)$$

Now select the matrix Q to be the transformation which diagonalizes $G(k)\overline{X}(k)\overline{X}^T(k)$. Then the diagonalized or modal form of Eq (68) is shown in Eq (69). The signal flow graph corresponding to Eq (69) is presented in Figure 10.

$$\overline{h}'(k+1) = \overline{h}'(k) + Q^{-1}G(k)\overline{X}(k)e(k) - \Lambda(k)\overline{h}'(k) \qquad (69a)$$

$$= [I - \Lambda(k)]\overline{h}'(k) + Q^{-1}G(k)\overline{X}(k)e(k) \qquad (69b)$$

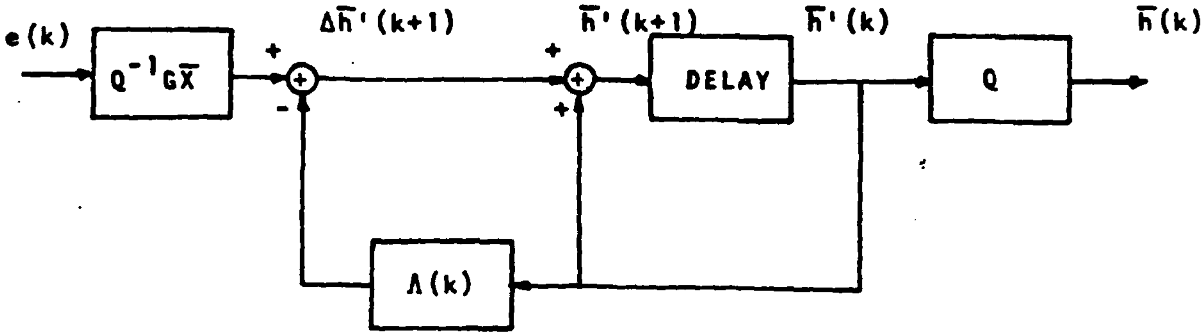


FIGURE 10. SIGNAL FLOW GRAPH FOR MISALIGNMENT VECTOR IN DIAGONALIZED COORDINATE SYSTEM.

We note that in the modal form of the difference equation, each coordinate (or component) of the misalignment vector operates independently. A signal flow graph of the m-th component in the modal form of the difference equation is shown in Figure 11.

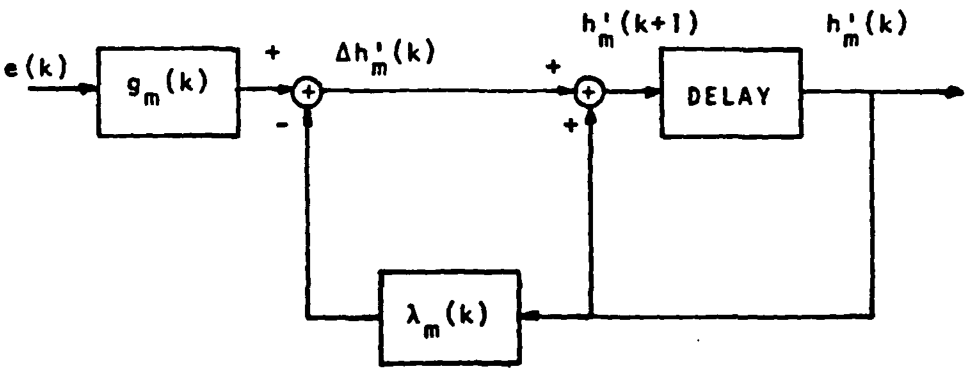


FIGURE 11. SIGNAL FLOW GRAPH FOR M-TH MODAL COORDINATE OF MISADJUSTMENT VECTOR

The difference equation corresponding to the separate modal coordinates is shown in Eq (70).

$$h'_m(k+1) = [1 - \lambda_m(k)]h'_m(k) + g_m(k)e(k) \tag{70}$$

Stability of the modal coordinates is assured if the homogeneous system Eq (71) is non increasing.

$$h'(k+1) = [1 - \lambda_m(k)]h'_m(k) \tag{71}$$

The homogeneous system is non increasing if the magnitude of the contraction constant $K_m(k)$, defined in Eq (72), is bounded by unity.

$$|K_m(k)| = |1 - \lambda_m(k)| \leq 1 \tag{72}$$

It now remains to examine the eigenvalues of $G(k)\overline{X}(k)\overline{X}^T(k)$, which are identical with the eigenvalues of $Q^{-1}G(k)\overline{X}(k)\overline{X}^T(k)Q$. This is show in expanded form in Eq (73).

$$\frac{B(k)\overline{X}(k)\overline{X}^T(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} \tag{73}$$

As shown, Eq (73) is a dyadic operator. It is a singular matrix with eigenvalues identified in Eq (74).

$$\lambda_1 = \frac{\overline{X}^T(k)B(k)\overline{X}(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)} \tag{74a}$$

$$= \frac{TR[B(k)\overline{X}(k)\overline{X}^T(k)]}{1 + TR[B(k)\overline{X}(k)\overline{X}^T(k)]} = \frac{C_1}{1+C_1} \tag{74b}$$

$$\lambda_2 = \lambda_3 = \lambda_4 = \cdot \cdot \cdot \cdot = \lambda_N = 0 \tag{74c}$$

Then for each iteration, the contraction constant $K_m(k)$ of Eq (72) is simply that shown in Eq (75).

$$K_m(k) = [1 - \lambda_m(k)] \tag{75a}$$

$$= \frac{1}{1 + TR[B(k)\overline{X}(k)\overline{X}^T(k)} \quad ; m = 1 \tag{75b}$$

$$= 1 \quad ; m \neq 1 \tag{75c}$$

Thus as a dyadic operator, the contraction constant is strictly non increasing in all coordinates if Eq (76) holds true.

$$TR[B(k)\overline{X}(k)\overline{X}^T(k)] \geq 0 \tag{76}$$

Since $B(k)\overline{X}(k)\overline{X}^T(k)$ has only one non zero eigenvalue, and since $TR[B(k)\overline{X}(k)\overline{X}^T(k)]$ is equal to the sum of the eigenvalues, then Eq (76) is equivalent to the requirement that the single eigenvalue by non negative. Or equivalently to requiring $B(k)\overline{X}(k)\overline{X}^T(k)$ be positive semidefinite. This in turn is equivalent to requiring that $<\overline{X}^T(k),B(k)\overline{X}(k)\geq 0$. From this we conclude that $B(k)$ must be positive definite. On the other hand, if we require at each iteration of the algorithm, that the contraction constant be strictly less than unity, then $B(k)$ must be positive definite.

Now rather than pursue the time varying active direction of the algorithm, we will examine the average of the active directions. We do so by applying the expectation operator to Eq (73) thus converting from a dyadic operator to a full rank operator. This is indicated in Eq (77).

$$E\left\{\frac{B(k)\overline{X}(k)\overline{X}^T(k)}{1 + \overline{X}^T(k)B(k)\overline{X}(k)}\right\} \tag{77a}$$

This can be expanded in an approximate form as indicated in Eq (77b).

$$\frac{E\{B(k)\overline{X}(k)\overline{X}^T(k)\}}{E\{1 + \overline{X}^T(k)B(k)\overline{X}(k)\}} \tag{77b}$$

Assuming that the matrix in Eq (77) has full rank, then the diagonalized form of $E\{B(k)\overline{X}(k)\overline{X}^T(k)\}$ has eigenvalues C_1 through C_N. The scaled eigenvalues of Eq (77b) are shown in Eq (78)

$$\lambda_m = \frac{c_m}{TR[\overline{X}^T(k)B(k)\overline{X}(k)]} \tag{78a}$$

$$= \frac{c_m}{\sum_{n=1}^{N} c_n} \tag{78b}$$

The homogeneous difference equation for the average misadjustment modal system is shown in Eq (79).

$$h'_m(k+1) = [1 - \lambda_m]\, h'_m(k) \tag{79a}$$

$$= \left[1 - \left(\frac{c_m}{\sum_{n=1}^{N} c_n}\right) h'_m(k)\right] \tag{79b}$$

The contraction constant $K_m(k)$ for this system is non increasing if Eq (80) holds true.

$$|K_m(k)| = \left[1 - \frac{c_m}{\sum_{n=1}^{N} c_n}\right] \leq 1 \tag{80a}$$

$$= \left[\frac{\sum_{\substack{n=1 \\ n \neq m}}^{N} c_n}{\sum_{n=1}^{N} c_n}\right] \leq 1 \tag{80b}$$

$$= \frac{TR[\overline{X}^T(k)B(k)\overline{X}(k)] - c_m}{TR[\overline{X}^T(k)B(k)\overline{X}(k)]} \leq 1 \tag{80c}$$

A sufficient condition for Eq (80) to hold true is that the eigenvalues C_m satisfy Eq (81).

$$C_m \geq 0 \tag{81}$$

Equation (81) is equivalent to requiring that the matrix $E\{B(k)\overline{X}(k)\overline{X}^T(k)\}$ be positive semidefinite. If the matrix $E\{B(k)\overline{X}(k)\overline{X}^T(k)\}$ is of full rank, (we assume it is) then Eq (81) must be satisfied with strict inequality which means the matrix $E\{B(k)\overline{X}(k)\overline{X}^T(k)\}$ is of course positive semidefinite.

X. RATES OF CONVERGENCE OF ALGORITHM.

Let us now examine the manner in which the C_m's, the eigenvalues of $E\{B(k)\overline{X}(k)\overline{X}^T(k)\}$, effect the transient behavior of the adaptive filters. Let us first consider the case in which all of the eigenvalues C_m are all approximately the same value or size. Then the contraction constants $K_m(k)$ are approximately of the form shown in Eq (82).

$$K_m(k) \approx \frac{\left[\sum_n C_n\right] - C_m}{\sum_n C_n} \tag{82a}$$

$$\approx \frac{N-1}{N} = 1 - \frac{1}{N} \tag{82b}$$

We note that N is the dimension of the non recursive filter and that the average transient time evolves over the time index "k" according to Eq (83).

$$\Delta h_m'(k) = \Delta h_m'(0) \; [K_m]^k \tag{83}$$

The relative change in a given weight misalignment coordinate that can occur in a single iteration is simply;

$$\frac{\Delta h_m'(k+1) - \Delta h_m'(k)}{\Delta h_m'(k)} \tag{84a}$$

$$= \frac{[k_m]^{(k+1)} - [K_m]^k}{[K_m]^k} \quad (84b)$$

$$= [K_m - 1] \quad (84c)$$

The relative change in the weight vector per iteration is of course proportional to the slope of the transient. Greater slopes imply faster transients, and smaller slopes imply slower transients. Substituting Eq (82b) into Eq (84c), we find the relative change per iteration to be;

$$\text{Relative change} = [-\frac{1}{N}] \quad (85)$$

If the transient were to continue at this slope, the transient terms would go to zero in an interval of precisely N iteration. This interval is the equivalent time constant of the iteration on the weight vector. In fact the transient essentially runs for four time constants. Note that the time constant for the equal size eigenvalue problem is the same as the filter length. Thus if the information matrix can hold the eigenvalues to near the same values, the transient will run for approximately four filter lengths. We also conclude that longer filters, while exhibiting smaller mean square error, have longer transient intervals,

Now let us consider the case where there is a large spread of the eigenvalues C_m. In particular, let us consider the case for which all but one of the eigenvalues is of the same size. The one exception is either larger C_{max} or smaller C_{min} by a factor β. Then the contraction constant $K_m(k)$ is of the form shown in Eq (86).

$$K_{m(\beta)} = \frac{\left[\sum_n C_n\right] - C_{max/min}}{\sum_n C_n} = \frac{[(N-1)+\beta] - \beta}{[(N-1)+\beta]} \quad (86a)$$

$$= 1 - \frac{\beta}{[(N-1)+\beta]} \quad (86b)$$

$$K_{m(1)} = \frac{[(N-1)+\beta] - 1}{[(N-1)+\beta]} \tag{86c}$$

$$= 1 - \frac{1}{[(N-1)+\beta]} \tag{86d}$$

The equivalent time constants are approximately $[\frac{N}{\beta} + 1]$ for the exceptional eigenvalue and approximately $[N+\beta]$ for the remaining eigenvalues. If β is much smaller than unity, indicating a single small eigenvalue, the corresponding time constant becomes very large. The remaining time constants are essentially unchanged. On the other hand, if β is significantly larger than unity, indicating a single large eigenvalue, the corresponding time constant becomes slightly smaller. The remaining time constants increase from approximately "N" to "N+β". Note that either way, if there is an exceptionally large or an exceptionally small eigenvalue, there is an increase in at least one time constant. If only one time constant increases, the total convergence time must increase. We note that any mix of small and large eigenvalues would lead to the same conclusion. Thus if there is a large spread of eigenvalues, the algorithm will exhibit long transient times. The function of the minimum variance or fading memory information matrices is to force nearly constant eigenvalues and thus ensure short convergence times. Note that the convergence rates are dependent upon the ratio of the eigenvalues and not their absolute sizes.

DATE

FILME

2-83